AI-Enabled IoT Air Quality Monitoring

Krushabh Roy[1], Hasnain Patil[2], Wahid Shaikh[3]
[1,2,3]*MIT School of Computing MIT ADT University Pune, India*

***Abstract*—Growing concerns about indoor air pollution have highlighted the need for intelligent, continuous monitoring systems capable of both detecting and anticipating harmful conditions. This paper describes a modular, AI-integrated air quality monitoring platform that collects environmental data—including PM2.5 particulate concentration, temperature, relative humidity, and gas levels—from physical or simulated sensors and transmits it to AWS IoT Core via the MQTT protocol. A backend built on FastAPI processes incoming readings, persists them in a structured database, and relays updates to client interfaces through persistent WebSocket connections.**

The analytical core of the system is a deep learning model based on LSTM/GRU architecture, trained to forecast near-future PM2.5 levels from historical time-series data. These forecasts feed into a rule-based reasoning layer that classifies current and predicted air quality against established health thresholds, issuing context-aware recommendations and early-warning alerts before conditions deteriorate. A Next.js frontend presents all of this through a responsive, real-time dashboard designed to be interpretable by users without technical expertise.

By unifying IoT communication, cloud infrastructure, machine learning, and interactive visualization into a single operational pipeline, the system advances beyond conventional reactive monitoring toward genuine predictive environmental management. Experimental evaluation confirms reliable real-time transmission latencies below 500 ms and PM2.5 forecasting accuracy in the 85–90% range, validating the architecture as a practical foundation for future smart-environment and precision-health applications.

***Index Terms*—Air Quality Monitoring; IoT; PM2.5 Prediction; MQTT Protocol; AWS IoT Core; FastAPI; Next.js; Machine Learning; LSTM; Real-Time Analytics; WebSockets**

I. INTRODUCTION

The quality of the air we breathe indoors is a frequently underestimated determinant of human health, cognitive performance, and general well-being. As cities grow denser and buildings more airtight, indoor pollutants accumulate in ways that conventional sensing approaches are poorly equipped to detect or address. Occupants often remain unaware of degrading conditions until visible or physical symptoms appear—by which point sustained exposure may have already caused measurable harm. The monitoring tools historically available were either expensive fixed installations suited to industrial environments or basic consumer devices with little analytical depth, leaving a significant capability gap for everyday indoor settings.

Advances in IoT hardware, cloud communication protocols, and machine learning have substantially changed what is achievable in this space. Lightweight sensors can now stream environmental readings continuously to cloud endpoints at minimal cost, while sequence-based neural networks have matured to the point where reliable short-term forecasting of air quality metrics is computationally tractable. The challenge that remains is not capability in isolation but integration: most existing systems solve one part of the problem well while leaving others unaddressed. A system that collects data but cannot visualize it helpfully, or one that visualizes data but cannot anticipate where conditions are heading, delivers only a fraction of the value that the underlying technology could provide.

This work responds to that integration challenge by presenting a complete, end-to-end air quality monitoring architecture that spans data acquisition, cloud transmission, backend processing, predictive analysis, and interactive visualization within a single cohesive design. The system captures four environmental parameters—PM2.5 particle concentration, ambient temperature, relative humidity, and gas levels—and conveys them to AWS IoT Core via MQTT. A FastAPI backend receives, processes, and stores these readings, simultaneously pushing updates to a Next.js frontend through WebSocket connections. A trained LSTM/GRU model runs continuously within this pipeline, generating near-term PM2.5 forecasts that, alongside live sensor values, drive a context-aware health advisory layer. The result is a platform that does not merely report conditions but actively anticipates them.

II. PROBLEM STATEMENT

Despite the abundance of platforms offering some form of air quality data—government monitoring networks, commercial sensor products, and web-based reporting services—a fundamental usability gap persists. Most of these tools are designed around a passive information model: they record what has occurred and display it after the fact, typically with processing delays that reduce the practical value of the data for time-sensitive decisions. They also tend to be monolithic in scope, providing coverage at the city or district level rather than reflecting the conditions within a specific room or building where an individual actually spends their time.

Several interconnected deficiencies characterize the current landscape:

- The absence of real-time, continuous monitoring of key indoor parameters including PM2.5, temperature, humidity, and gas concentrations.
- Reliance on polling-based data delivery rather than event-driven architectures that propagate updates as soon as new data arrives.
- Limited or nonexistent predictive capability, which confines systems to describing present or past conditions rather than helping users prepare for what is approaching.
- Health guidance that is generic rather than derived from a user's actual measured and forecasted environment.
- Interfaces designed with technical audiences in mind, creating barriers for the broader public who would benefit most from accessible environmental data.

Addressing these limitations requires a system designed from the ground up around continuous operation, intelligent analysis, and user-centered presentation—the three properties that this work sets out to combine.

III. GOALS

A. Primary Objective

The central aim of this research was to develop a practically deployable indoor air quality monitoring system that transcends the limitations of existing reactive approaches. Rather than building another data-collection tool, the objective was to create an intelligent platform that understands the temporal dynamics of indoor pollution and translates that understanding into timely, actionable guidance for users. This required not only accurate sensing and reliable transmission but also a predictive engine capable of forecasting PM2.5 concentrations and a reasoning layer capable of connecting those forecasts to concrete health recommendations.

The guiding design principle throughout was that technical sophistication should remain invisible to the end user. A system that surfaces insights in terms a non-specialist cannot interpret has not fully completed its job. The interface was therefore treated as a first-class concern rather than an afterthought, with the goal of making environmental health status immediately legible to anyone who interacts with it.

B. Secondary Objectives

Real-Time Data Acquisition

Establishing a steady, low-latency data pipeline was treated as a foundational requirement. The system, whether drawing from physical hardware or a software simulation, captures PM2.5 levels, temperature, humidity, and gas concentrations at five-second intervals and transmits them through an MQTT channel into AWS IoT Core, ensuring that the monitoring loop remains tight enough to detect meaningful environmental changes as they develop.

Data Processing and Persistence

Incoming sensor streams carry little inherent meaning without a robust processing layer behind them. The FastAPI backend handles validation, parsing, and normalization of arriving data before writing records to a structured SQLite database, maintaining a clean and queryable historical archive that supports both trend analysis and model training.

Predictive Analysis

Forecasting was treated as a core system function rather than an optional enhancement. An LSTM/GRU model trained on historical time-series data runs continuously within the pipeline, consuming recent readings of PM2.5, temperature, and humidity to project likely concentration levels over the next hour. This forward-looking capability reframes the system from a passive observer into an active early-warning instrument.

Health Insight Generation

Raw forecasts become meaningful only when they are connected to outcomes that users can understand and act on. A rule-based reasoning module compares current and predicted air quality values against WHO-aligned health thresholds, categorizing conditions and generating specific recommendations—such as ventilation adjustments or activity modifications—before thresholds are breached rather than after.

Real-Time Visualization

The Next.js frontend delivers live charts that render PM2.5 readings alongside model-generated forecasts, giving users an immediate visual sense of both where conditions stand and where they appear to be heading. Companion graphs cover temperature, humidity, and gas levels, while a paginated historical table provides access to the full data record.

Accessible User Interface

Interface design decisions were consistently evaluated against the question of whether a non-technical user could interpret the output and take appropriate action. This guided choices around layout, terminology, visual hierarchy, and alert presentation, with the goal of lowering the barrier to effective use without sacrificing the informational depth that more engaged users might seek.

IV. LITERATURE SURVEY

The trajectory of research in IoT-based environmental monitoring reflects a discipline that has grown technically sophisticated while remaining fragmented in terms of end-to-end system design. Work in this space has progressed along three largely parallel tracks—hardware and sensing infrastructure, predictive modeling, and user-facing applications—without consistently bringing all three together into a unified architecture. Reviewing contributions across these tracks clarifies both what has been accomplished and what remains to be integrated.

Foundational infrastructure work by Zanella et al. [1] demonstrated that distributed sensor nodes could be organized into city-scale networks capable of continuous passive data collection. The architecture they proposed was technically sound and pointed toward what city-wide environmental sensing might eventually look like, but it treated data collection as the endpoint rather than the starting point for analysis. Similarly, Kumar et al. [2] prioritized hardware accessibility by designing a low-cost Raspberry Pi-based sensing platform, widening the potential deployment base considerably. The limitation was that affordability was achieved at the expense of analytical depth—the system delivered raw readings without any mechanism for interpretation or forecasting.

Gubbi et al. [3] articulated an influential conceptual framework for interconnected IoT environments, addressing scalability and interoperability concerns at the architectural level. While their vision has shaped subsequent infrastructure designs, it deliberately left questions of data analysis and intelligent response outside its scope. WHO air quality guidelines [4] have provided the field with a shared reference standard for pollutant thresholds, and they remain the most widely used basis for health categorization—though their application in most deployed systems is limited to static comparisons against already-observed values rather than against anticipated future conditions.

Predictive modeling became a serious research thread following the demonstration by Hochreiter and Schmidhuber [5] that Long Short-Term Memory networks could reliably learn dependencies across extended time sequences. Zhang et al. [6] applied this architecture to the specific problem of PM2.5 forecasting, later incorporating GRU variants, and reported accuracy results that validated the approach for environmental time-series data. Their work, however, was conducted in an offline experimental context: models were trained and evaluated against historical datasets without any connection to a live sensor pipeline. Li et al. [7] reached similar conclusions through deep learning experimentation, again producing compelling accuracy figures but stopping short of operational integration.

On the infrastructure side, MQTT has emerged as the preferred transport protocol for constrained IoT devices owing to its low overhead and publishsubscribe model, with AWS IoT Core [8] serving as one of the more widely adopted cloud destinations for MQTT-transmitted sensor data. These components handle the data movement layer efficiently but carry no analytical function of their own—data that arrives is stored or forwarded without being interpreted. FastAPI [9] has attracted attention as a high-performance backend framework well suited to data-intensive web services, though implementations that leverage its capabilities often default to request-response patterns rather than the persistent WebSocket connections that would enable genuinely continuous updates.

Singh et al. [10] brought visualization into focus with a cloud-hosted dashboard that translated sensor readings into interpretable charts, marking a meaningful step toward making air quality data accessible to general users. The system suffered from processing latency that undermined its real-time claims and lacked any predictive component. Verma et al. [11] extended accessibility further through mobile delivery, placing monitoring within constant reach but without the ability to look ahead at where conditions were heading. Abdel-Salam et al. [12] addressed the reliability and energy efficiency of wireless sensor networks at the hardware level, contributing knowledge important to any physical deployment—though their work did not engage with how collected data should be presented or responded to by end users. Khedo et al. [13] proposed a distributed multi-node pollution sensing architecture, sensibly distributing sensing load, but without building any form of automated reasoning or user guidance on top of it.

More complex predictive approaches include the hybrid model architectures developed by Jiang et al. [14], which combined multiple model types to improve accuracy at the cost of computational complexity that creates practical obstacles in latency-sensitive deployment contexts. Yu et al. [15] pursued deep learning-based pollutant forecasting and produced technically strong results, but their work was framed as an analytical tool rather than a deployable system component. Recent efforts by Sharma et al. [16] and Patel et al. [17] have moved meaningfully toward integrated IoT-cloud monitoring platforms that lower barriers for general users, yet both operate in fundamentally reactive mode—surfacing information about conditions that have already materialized, without predictive capability or proactive health guidance.

Taken together, this body of work reveals a consistent pattern: individual components of an intelligent air quality monitoring system have been demonstrated and validated in isolation, but their integration into a complete, operational pipeline that spans from sensing through prediction to user action remains largely unexplored.

V. DIFFERENTIATION FROM PRIOR WORK

The most significant distinction between the system presented here and prior contributions lies not in any single technical innovation but in the coherence of the overall architecture. Where existing work tends to optimize one dimension—sensing fidelity, predictive accuracy, or interface usability—while leaving others underdeveloped, this system treats all three as equally essential components of a single operational design. Examining specific prior contributions makes the nature of these gaps concrete.

Zanella et al. [1] demonstrated that distributed IoT sensing could work at scale, but the data those networks collected remained largely inert. There was no analysis layer, no forecasting mechanism, and no interface oriented toward individual users. The system described in this paper takes the same underlying premise—distributed environmental sensing—and extends it through a full processing pipeline that transforms raw readings into health guidance delivered in real time.

The affordability-focused sensing work of Kumar et al. [2] widened access to air quality measurement but did not address what users should do with the numbers they received. Accessible hardware is a necessary condition for broad deployment, not a sufficient one. Here, the sensing layer is treated as an input to an analytical system designed to make its outputs immediately interpretable, pairing data collection with the reasoning infrastructure needed to give that data practical meaning.

The architectural vision articulated by Gubbi et al. [3] correctly identified scalability and interconnection as design priorities for large-scale IoT deployment. What their framework deliberately set aside—the intelligence layer that would make such networks useful beyond data warehousing—is precisely what the predictive and advisory components of this system provide. The WHO thresholds codified in [4] appear in most monitoring systems as fixed comparison points applied to static snapshots; the approach taken here applies them dynamically against both live sensor readings and forecast values, so that guidance reflects not just current conditions but those that are developing.

The LSTM/GRU forecasting work of Zhang et al. [6] and Li et al. [7] provided rigorous validation that sequence-based neural networks can predict PM2.5 concentrations with meaningful accuracy. The limitation in both cases was the absence of an operational context: models trained on historical data and evaluated offline do not automatically become useful to someone monitoring a room in real time. In this system, the forecasting model is embedded within a live data pipeline, receiving continuously updated inputs and generating predictions that are surfaced to users as conditions evolve.

The IoT infrastructure components—MQTT transmission to AWS IoT Core [8] and FastAPI backend processing [9]—are used by this system in ways that extend beyond their typical deployment patterns. Rather than treating AWS IoT Core merely as a data sink, the architecture connects it to a processing layer that performs active inference. Rather than using FastAPI for standard request-response API delivery, the system leverages WebSocket channels to maintain persistent connections that push updates to the frontend as new data arrives, eliminating the polling delays that degrade the real-time experience in many comparable implementations.

The visualization work of Singh et al. [10] and Verma et al. [11] moved meaningfully toward making air quality data accessible to general users, but both were constrained to showing what has already happened. Displaying current or recent conditions is useful; displaying what conditions are likely to look like in the next hour, alongside recommendations for action, is substantively more useful. The Next.js dashboard in this system is designed around this distinction, presenting historical data, live readings, and model forecasts together so that users can form a complete picture of their environment's past, present, and near future.

The higher-accuracy hybrid modeling of Jiang et al. [14] and the deep learning forecasting of Yu et al. [15] represent the upper end of predictive capability in this domain. Their trade-off—accuracy at the cost of computational speed—is acceptable for research analysis but problematic for a system that needs to generate and display predictions continuously. The approach taken here prioritizes inference speed sufficient for real-time operation, accepting that a well-optimized LSTM/GRU model delivers accuracy well within the range needed for health monitoring purposes.

Finally, the integrated IoT-cloud monitoring efforts of Sharma et al. [16] and Patel et al. [17] demonstrate genuine progress in combining hardware sensing with cloud-based visualization. Both systems, however, remain fundamentally reactive: they report on conditions that have already occurred. The system presented here is designed to operate one step ahead, issuing alerts before thresholds are crossed and framing health guidance around anticipated conditions—a qualitative difference in how the system serves its users.

VI. METHODOLOGY

The system is organized into six functionally distinct modules, each responsible for a specific stage of the data and inference pipeline. These modules are designed to interact through well-defined interfaces, allowing individual components to be upgraded or replaced without disrupting the rest of the system.

A. User Interface Module

The frontend serves as the user-facing layer of the system and was developed using Next.js to support fast client-side rendering and fluid navigation. Upon authentication, users access a centralized dashboard organized into four primary views: Overview, Trends, Insights, and Historical Data. Each view is designed to surface a different temporal and contextual lens on the monitored environment.

Two conditions are verified before granting dashboard access:

- User identity is confirmed through a JWT-based authentication mechanism, preventing unauthorized session access.
- Active session state is validated on page load, with the previous system state automatically restored to ensure continuity across browser refreshes.

B. Data Acquisition Module

This module manages the collection of environmental readings from sensor sources, either physical or simulated:

- Four parameters are captured: PM2.5 particulate concentration, ambient temperature, relative humidity, and gas levels.
- Readings are generated or sampled at five-second intervals to maintain a continuous, uninterrupted data stream.
- Data is transmitted to AWS IoT Core via the MQTT protocol, using TLS certificate-based authentication to secure the communication channel against unauthorized access.

C. Data Ingestion and Processing Module

Once sensor data arrives at the cloud endpoint, this module handles its preparation for storage and downstream processing:

- The FastAPI backend maintains a persistent subscription to the relevant MQTT topics, receiving and parsing incoming messages as they arrive.
- Each data packet undergoes validation and structural normalization before being accepted for storage.
- Processed records are written to a SQLite database, building the historical archive that underpins both trend analysis and model inference.
- Simultaneously, each new record is broadcast to all active frontend clients via WebSocket connections, keeping dashboard displays current without polling.

D. Predictive Analysis Module

This module is responsible for generating near-term PM2.5 forecasts from the incoming time-series data:

- Historical records are retrieved, cleaned, and normalized using a standard scaler prior to model input.
- A pre-trained deep learning model combining LSTM and GRU layers handles sequential forecasting, having been trained specifically on environmental time-series data.
- Input features consist of recent sequences of PM2.5 readings alongside corresponding temperature and humidity values.
- Model output is a projected PM2.5 concentration level for the upcoming hour, which is merged into the live data stream and surfaced alongside current sensor readings for direct comparison.

E. Health Insights and Alert Module

This module translates raw measurements and forecasts into guidance that users can act on:

- Current air quality values are classified against established thresholds—Good, Moderate, Unhealthy for Sensitive Groups, Unhealthy, Very Unhealthy, and Hazardous—aligned with WHO guidelines.
- A comparative assessment runs between live sensor values and model-generated forecasts to identify trends and trajectory.
- When projected values indicate an imminent threshold crossing, the module issues early-warning alerts with enough lead time for users to take preventive action.
- Health recommendations and ventilation guidance are generated dynamically based on prevailing and anticipated conditions.

F. Visualization and Result Display Module

This module is responsible for presenting all system outputs in an accessible and continuously updated interface:

- Live PM2.5 charts render current sensor readings alongside model forecasts, giving users a simultaneous view of present conditions and near-future projections.
- Companion interactive graphs display temperature, humidity, and gas concentration trends over selectable time windows.
- A paginated, searchable historical table provides full access to the stored record of past readings.
- All displays update through the persistent WebSocket connection, eliminating the latency associated with periodic polling.
- Alerts and health advisories are surfaced through prominent visual indicators, ensuring that critical information is not overlooked regardless of which dashboard view is currently active.

VII. SYSTEM ARCHITECTURE FLOWCHART

Figure 1 illustrates the end-to-end data flow through the system, from sensor data generation through cloud transmission, backend processing, predictive inference, health evaluation, and final visualization. The pipeline is designed to operate continuously with minimal latency at each transition point.

[Figure 1: System Architecture Flowchart — see original submission]

VIII. RESULTS AND DISCUSSION

A. System Performance

Real-time transmission performance was evaluated by measuring end-to-end latency from sensor data generation to dashboard display. Under continuous operation, the MQTT communication channel through AWS IoT Core consistently delivered messages within 500 ms, with the majority of transmissions completing considerably faster. All four monitored parameters—PM2.5, temperature, humidity, and gas levels—propagated to the frontend without perceptible delay, and the WebSocket delivery mechanism eliminated the additional latency that polling-based alternatives would have introduced.

Forecasting performance was evaluated against a held-out portion of the historical dataset. The LSTM/GRU model achieved PM2.5 prediction accuracy in the 85–90% range across test scenarios, demonstrating consistent ability to capture the temporal patterns that drive gradual environmental changes. Predictions were less reliable during episodes of rapid, unpredicted concentration change—a known limitation of autoregressive time-series models that rely on historical pattern continuity—though accuracy remained sufficient for the health advisory use case even in these cases.

B. Analytical Outcomes

The health insight generation layer functioned as intended across evaluated scenarios, correctly classifying air quality conditions and producing recommendations aligned with observed and projected values. When PM2.5 forecasts indicated an approaching threshold crossing, the module successfully generated early-warning alerts with sufficient lead time for users to initiate ventilation or other corrective measures before conditions worsened. The specificity of the recommendations—grounded in actual sensor readings rather than generic advisory text—was consistently noted as a distinguishing feature during informal evaluation.

C. Key Findings

The integration of predictive analytics into the monitoring workflow produced a qualitative change in how users engaged with the system. The availability of near-future projections encouraged proactive responses to deteriorating conditions, a behavioral shift that a purely reactive system cannot support. The pairing of FastAPI's high-throughput request handling with WebSocket-based data delivery kept the dashboard genuinely live in a way that periodic API polling cannot match—an important distinction in environments where air quality can shift meaningfully within minutes. The Next.js interface succeeded in presenting complex multi-dimensional data in a form that test users found navigable and interpretable without technical guidance.

D. Limitations

Four limitations of the current implementation merit explicit acknowledgment. First, the model's prediction accuracy degrades when pollution levels shift abruptly, as sudden spikes or drops fall outside the distributional range on which the model was trained. Second, the system operated on simulated sensor data rather than physical hardware throughout the evaluation period; while the simulation was designed to be representative, it cannot fully reproduce the noise characteristics and failure modes of real sensors. Third, SQLite's concurrency constraints and write throughput ceiling would present scalability challenges in multi-user or high-frequency deployment scenarios, necessitating a more capable database backend before production use at scale. Fourth, the system's real-time functionality is entirely contingent on network connectivity; any disruption to the MQTT channel or WebSocket connection interrupts data flow until the connection is re-established.

IX. FUTURE SCOPE

- Integration of physical sensor hardware to replace simulated data sources, enabling field validation under real-world environmental variability and sensor noise conditions.
- Exploration of advanced forecasting architectures—including Transformer-based sequence models and ensemble hybrid approaches—that may improve prediction accuracy for abrupt environmental transitions and longer forecast horizons.
- Redesign of the data persistence layer around a distributed database solution to support higher

concurrent user loads and larger historical datasets as system deployment scales.

- Development of personalized recommendation logic that incorporates user-specific factors such as location context, historical exposure patterns, and individual health profiles to move beyond generalized advisory guidance.
- Integration with mobile platforms and smart home infrastructure—including automated ventilation systems and air purifiers—to enable autonomous corrective responses rather than relying solely on user-initiated action.

X. CONCLUSION

This work demonstrates that the convergence of IoT sensing, cloud infrastructure, deep learning, and real-time visualization can be realized not merely as a research proposition but as a functional, deployable system architecture. The platform described here addresses a genuine gap in existing monitoring approaches by moving from reactive condition reporting to proactive environmental intelligence: it monitors continuously, forecasts reliably, and communicates its findings in a form that general users can understand and act on.

The technical evaluation confirmed that MQTT-based data transmission through AWS IoT Core sustains end-to-end latencies below 500 ms under operational conditions, and that the integrated LSTM/GRU forecasting model achieves PM2.5 prediction accuracy in the 85–90% range—sufficient for health advisory applications. The combination of FastAPI's backend throughput with WebSocket-based delivery ensures that the dashboard remains genuinely live rather than periodically refreshed, a distinction that matters when environmental conditions can shift within the span of minutes.

Beyond what the system accomplishes in its current form, the architecture is explicitly designed as a foundation rather than a final product. The modular structure allows individual components—the sensing layer, the prediction engine, the health reasoning module—to be upgraded independently as better approaches emerge. The trajectory is clear: smarter prediction models, physical sensor deployment, personalized advisory logic, and integration with autonomous environmental control systems. What this work establishes is that the foundation is sound, and the path forward is well-defined.

REFERENCES

[1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, “Internet of Things for Smart Cities,” IEEE Internet of Things Journal, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[2] A. Kumar, I. P. Singh, and S. K. Sud, “Energy Efficient Air Quality Monitoring System Based on IoT Using Raspberry Pi,” Procedia Computer Science, vol. 132, pp. 1343–1352, 2018.

[3] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions,” Future Generation Computer Systems, vol. 29, no. 7, pp. 1645–1660, Sep. 2013.

[4] World Health Organization, “WHO Global Air Quality Guidelines,” 2021. [Online]. Available: https://www.who.int

[5] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[6] Y. Zhang, Q. Chen, and Z. Yang, “A Deep Learning Approach for Air Quality Prediction Using LSTM Networks,” IEEE Access, vol. 6, pp. 73724–73732, 2018.

[7] X. Li, L. Peng, Y. Hu, J. Shao, and T. Chi, “Deep Learning Architecture for Air Quality Predictions,” Environmental Science and Pollution Research, vol. 23, pp. 22408–22417, 2016.

[8] Amazon Web Services, “AWS IoT Core Developer Guide,” 2023. [Online]. Available: https://docs.aws.amazon.com/iot/

[9] S. Ramírez, “FastAPI Documentation,” 2023. [Online]. Available: https://fastapi.tiangolo.com/

[10] R. Singh and P. Sharma, “Cloud-Based Air Quality Monitoring System Using IoT,” International Journal of Environmental Science, vol. 10, no. 2, pp. 45–52, 2019.

[11] S. Verma and A. Kumar, “Real-Time Air Quality Monitoring System Using IoT and Mobile Applications,” International Journal of Engineering Research & Technology, vol. 9, no. 5, pp. 120–125, 2020.

[12] T. Abdel-Salam, A. M. M. El-Sayed, and M. A. Abdel-Gawad, “Wireless Sensor Networks for

Environmental Monitoring Applications," IEEE Sensors Journal, vol. 14, no. 8, pp. 2348–2355, Aug. 2014.

[13] M. Khedo, R. Perseedoss, and A. Mungur, "A Wireless Sensor Network Air Pollution Monitoring System," International Journal of Wireless & Mobile Networks, vol. 2, no. 2, pp. 31–45, 2010.

[14] W. Jiang, Y. Wang, and X. Zhang, "Hybrid Machine Learning Model for Air Quality Prediction," Atmospheric Environment, vol. 199, pp. 241–250, Feb. 2019.

[15] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu, "Deep Learning: A Generic Approach for Air Quality Prediction," IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 6, pp. 1107–1120, Jun. 2019.

[16] P. Sharma and V. Gupta, "IoT-Based Smart Air Quality Monitoring System with Cloud Integration," Journal of Ambient Intelligence and Humanized Computing, vol. 12, pp. 1235–1245, 2021.

[17] K. Patel and M. Shah, "Real-Time Environmental Monitoring Using IoT and Cloud Computing," International Journal of Computer Applications, vol. 176, no. 32, pp. 1–6, 2020.